

AD-A072 768

ELECTRONICS ENGINEERING GROUP (1842ND) SCOTT AFB IL
LINE ACCESS TEST SYSTEM (LATS). (U)

F/G 9/2

UNCLASSIFIED

JUL 79 M VERSTEGEN, J MCCORMACK
1842 EEG/EETE/TR-79-12

NL

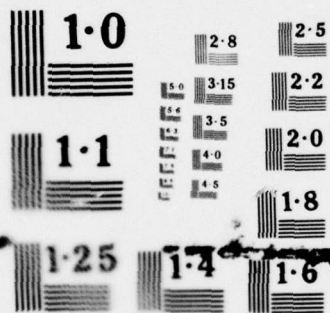
| OF |
AD
A072768



END
DATE
FILMED

9-79

DDC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART



AD A072768

DDC FILE COPY

LEVEL II

1842 EEG/EETE TR 79-12

AFCS TECHNICAL REPORT

LINE ACCESS TEST SYSTEMS (LATS)

AFCS SYSTEMS APPLICATION FACILITY
1842 ELECTRONICS ENGINEERING GROUP (AFCS)
SCOTT AIR FORCE BASE, ILLINOIS 62225

DDC
RECEIVED
AUG 14 1979
D

15 JULY 1979

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

79 08 13 091

409 646

1842 ELECTRONICS ENGINEERING GROUP

MISSION

The 1842 Electronics Engineering Group (EEG) has the mission to provide communications-electronics-meteorological (CEM) systems engineering and consultive engineering support for AFCS. In this respect, 1842 EEG responsibilities include: Developing engineering and installation standards for use in planning, programming, procuring, engineering, installing and testing CEM systems, facilities and equipment; performing systems engineering of CEM requirements that must operate as a system or in a system environment; operating a specialized Digital Network System Facility to analyze and evaluate new digital technology for application to the Defense Communications System (DCS) and other special purpose systems; operating a facility to prototype systems and equipment configurations to check out and validate engineering-installation standards and new installation techniques; providing consultive CEM engineering assistance to HQ AFCS, AFCS Areas, MAJCOMS, DOD and other government agencies.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 1842 EEG/EETE / TR-79-12	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LINE ACCESS TEST SYSTEM (LATS)	5. TYPE OF REPORT & PERIOD COVERED Final rept.	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lt Mike Verstegen CIC John McCormack	8. CONTRACT OR GRANT NUMBER(s)	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
9. PERFORMING ORGANIZATION NAME AND ADDRESS 1842 EEG/EETEX Scott AFB, IL 62225	12. REPORT DATE 15 July 79	13. NUMBER OF PAGES 25
11. CONTROLLING OFFICE NAME AND ADDRESS 1842 EEG/EEIS Scott AFB, IL	15. SECURITY CLASS. (of this report) UNCLASSIFIED	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12-26p.	16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release. Distribution unlimited	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) N/A		
18. SUPPLEMENTARY NOTES This report documents work by CIC John McCormack during the Cadet Summer Research Project 1978		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Microprocessor Line Access Test Set		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This technical report discusses the Line Access Test System (LATS) which can perform matrix switching operations on four wire circuits. Each circuit accessed may be tested for D.C. voltage, resistance, A.C. voltage and capacitance. A microprocessor control implementation is also included with the necessary programs for operation.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409646

APPROVAL PAGE

This report has been reviewed and is approved for publication and distribution.



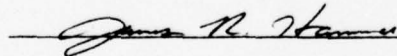
ZYGMUND J. BARA

Chief, Telecommunications Systems
Engineering Division



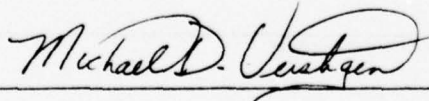
PHILIP E. GLENN, Capt, USAF

Chief, Systems Measurement and
Evaluation Branch



JAMES R. HAMMER, Capt, USAF

Technical Area Manager



MICHAEL D. VERSTEGEN, 2Lt, USAF

Electronics Engineer

ABSTRACT

This technical report discusses the Line Access Test System (LATS) which can perform matrix switching operations on four wire circuits. Each circuit accessed may be tested for D.C. voltage, resistance, A.C. voltage and capacitance. A microprocessor control implementation is also included with the necessary programs for operation.

TABLE OF CONTENTS

<u>Para</u>		<u>Page</u>
1.0	BACKGROUND	1
2.0	TECHNICAL DISCUSSION	1
2.1	LATS Description	1
2.2	LATS Interface	2
2.3	LATS Software	2
Appendix A	LATS CONTROLLER SOFTWARE LISTING	A-1 thru A-16

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	RELAYS IN ONE CIRCUIT IN MATRIX UNIT	4

1.0 BACKGROUND.

1.1 The Line Access Test System (LATS) was a joint Air Force Academy/1842 EEG project accomplished under the cadet summer research program in 1978. Cadet First Class John McCormack undertook the LATS project and completed all of the requirements in the six week span of the program.

1.2 To date, the LATS interface is complete in both hardware and software and has been put into service for specific test applications.

2.0 TECHNICAL DISCUSSION.

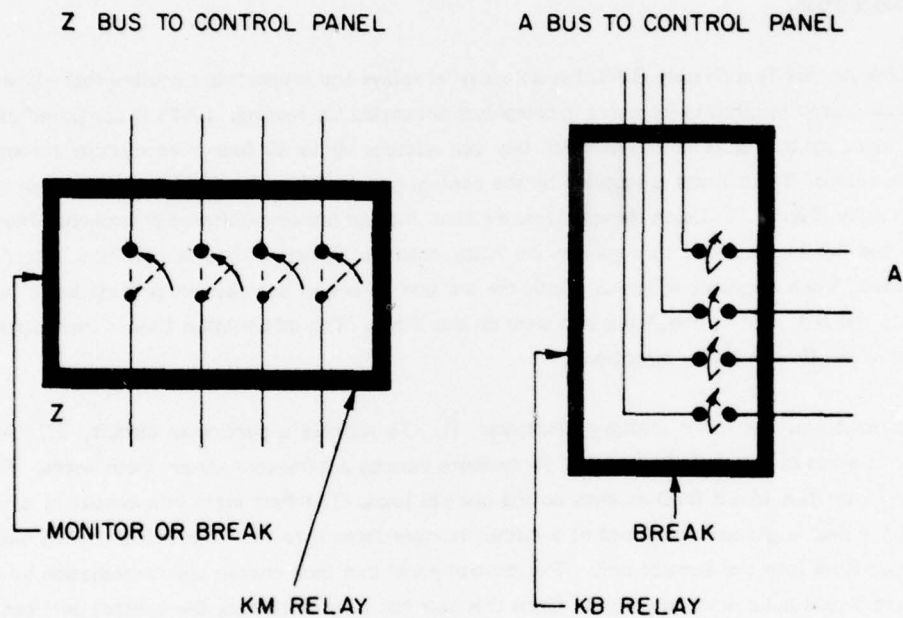
2.1 LATS Description.

2.1.1 The Line Access Test System (LATS) is an array of relays and supporting circuitry that allows a user to access individual four wire circuits (common in telephone networks) for testing. LATS is composed of a control panel and up to 60 matrix units or bays. Each bay can address up to 80 four-wire circuits through opening contacts on the relays. The address is supplied by the control panel. Each circuit is passed through two relays, a km and a kb relay (Figure 1). Using these relays, a circuit may be either monitored or broken. When a circuit is monitored, the input is shunted through the km relay onto the Z (west) data bus without interrupting the information flow. When a circuit is broken, both the km and kb relays are opened, putting both the east and west wires onto the bus, east on the A bus and west on the Z bus. The information flow is interrupted and the user may observe the flow in either direction.

2.1.2 The control unit has three primary functions: (1) To address a particular circuit, (2) To access a particular pair of wires in that circuit, and (3) To measure various parameters across those wires. To access a circuit, the LATS sends a 16-bit BCD address on the control lines. The first eight bits consist of a matrix unit address, while the next eight are composed of a circuit number from 0 to 79. The circuit, when monitored or broken, ties four lines into the control unit. The control panel can then choose any combination of these as a pair, or compare a particular wire to ground. Once this pair has been accessed, the control unit can test four parameters: D.C. voltage, resistance, A.C. voltage, and capacitance. The scales of these measurements are fixed, rather than floating, so that an off-scale or overflow reading causes the display to show "OF".

2.1.3 LATS is equipped with external jacks for attaching reliable measuring equipment to the accessed circuit. Eight lines (R, T, R1, T1, Rz, Tz, R1z, T1z) are brought out so that the pair selected is available for measurements by external equipment.

2.1.4 The control panel is equipped with a self test capability. In addition, there are two indicator lights. One indicates that the circuit was not accessed; this is the Fault (FLT) signal, which usually means that the selected bay was not present. The other indicator is the MIC (micro in control) which indicates that the LATS is being controlled by a computer. The interface consists of three 50-pin connectors on the back panel of the control unit connected to five 16-pin DIP ports on the Edumicro.



NOTE: When a break command is given, both relays switch their information to the two data buses. When a monitor is given, only the KM relay is switched causing Z to go onto the data bus; therefore, the circuit is not interrupted.

Figure 1. Relays in One Circuit in Matrix Unit

2.2 LATS Interface.

2.2.1 Edumicro is an Intel 8080A Based Micro Processor system developed by USAFA/DFEE. The basic Edumicro has onboard 2k of RAM (Random Access Memory), 8 input and 8 output ports, 1k of ROM (Read Only Memory) monitor and room for 1k of user ROM. The user ROM is filled by the LATS monitor when the Edmuicro is used to control the LATS.

2.2.2 The connections between the LATS and Edumicro are a straightforward process. Using the LATS interface specification, the control bits are grouped into eight-bit bytes which can be processed by the INTEL 8080A. The MIC is controlled by a toggle switch.

2.3 LATS SOFTWARE.

2.3.1 The approach to the software was to determine exactly what the LATS was capable of doing and to supply the necessary control sequence to accomplish it. For example, to call up a four wire circuit, the previous circuit must first be cleared by outputting the CCSD (clear circuit select data) bit. Next, the circuit must be addressed. The address is sixteen BCD bits, an eight-bit bay (or Matrix unit) address, and an eight-bit circuit address. The format for transferring this information is in a sequence of four 4-bit words (CKB) strobed by a fifth line (CKST) for a specified time interval.

2.3.2 The next step in accessing a circuit is making sure that the circuit that was supposed to be called was the one actually called. This is done by inputting BTCK (bay tens check), BUCK (bay units check), OTCK (circuit tens check), and OUCK (circuit units check), and comparing the input to what was put out. The circuit must then be broken or monitored by outputting the appropriate bit, delaying, and inputting the check bit. Under micro control, the break and monitor set bits do not clear automatically; so once set, these bits need not be reset or regenerated for every circuit. The user must input and check the FLT bit (corresponding to the indicator light) for an unaccessed circuit.

2.3.3 Now that a circuit has been accessed, it may be checked for various conditions by the LATS; but first, the user needs to select a pair of wires by outputting the MTC (meter configuration) bits and inputting the MTCK (meter configuration check) bits for a check. Once on a pair of wires, the user may select a particular function to check, such as D. C. potential, by putting the proper code on the RFB lines and strobing the range/function information with RFS (range-function strobe). Next, RFN must be inputted and checked to insure that the information put out was the information received.

2.3.4 To read the meter, the meter must first be turned on (MTON) and checked for being on (MTRK). Then the user must start the reading (MST) and wait for the EOC (end of conversion) input to signal that the reading is complete. The user must check the OVR bit (comparable to the "OF", overflow, display) to insure that the reading was within the range selected. If it was not within range, the range must be changed.

2.3.5 The computer is capable of checking all of the circuits for all parameters without the operator having to key it; so the software counts. It counts through all of the circuit functions in all of the bays. It also performs a zero test for proper meter operation and a grounding test for proper relay operation; this process is explained in Appendix A.

2.3.6 Perhaps the user may not have sixty bays present, or he may only want to test ten circuits instead of eighty. The software then compensates. The program expects several user-keyed inputs: the lower limit and upper limit of numbered bays to be tested; and, if only one bay is being tested, the lower and upper limits of circuits within that bay. The user also determines whether to break or monitor all circuits and whether or not he wishes to run a grounding test to check relay operation.

2.3.7 Because of the length of the LATS words (four-bits and the abundance of single-bit instructions) the Edumicro output information was combined into usable eight-bit status words and saved in memory. Status words allow the program to change a bit of output and remember the change every time the output port is used. In addition to status words, there is also a series of flag words; one to remember whether to break or monitor and another to signal when all testing has been done. If the user wants to count circuits after ground testing, the done flag (DNFLG) must be reset.

2.3.8 The program contains an error message routine which displays a particular one byte message which will indicate either FLT, overflow, transfer error, or an unacceptable reading of a grounded input. After a key is depressed, the display will tell the user where the error occurred. The user is then given the option of keying in an address and resuming calculations at that location.

2.3.9 The monitor is flexible enough for a user to access one circuit without counting to the next circuit, to access a single function or a single pair without scanning, or to use the counting routines and check every possible combination.

APPENDIX A

AFCS 8080 ASSEMBLER

LATS CONTROLLER

10:05 169/79

PAGE 1

LINE LOC B1 B2 B3

SOURCE LINE

```

1      NAM LATS CONTROLLER
2      ****
3      *
4      *
5      * THIS PROGRAM IS DESIGNED TO CONTROL THE LATS USING
6      * THE EDUMICRO COMPUTER. IT INITIALIZES THE SYSTEM
7      * BY STORING THE VALUES OF THE DESIRED BAYS AND
8      * CIRCUITS TO BE TESTED, AS WELL AS ZERO TESTING THE
9      * UNIT. IT IS FLEXIBLE FOR ANY FURTHER TESTING,
10     * CONTAINING SUBROUTINES FOR CONFIGURING AND READING
11     * THE LATS
12     *
13     *      VARIABLE LIST:
14     *      NAME      CONTENTS      ASSIGNMENT
15     * STPNTR      INITIAL VALUE STACK POINTER      0CDF
16     * BAY          0CE0
17     * TPBAY        0CE1
18     * CKT          0CE2
19     * TPCKT        0CE3
20     * MOT          METER OUTPUT      0CE4
21     * PNTR         POINTER TO FUNCTION TABLE      0CE5
22     * -            (2 BYTE)          0CE6
23     * EC           ERROR COUNT       0CE7
24     * STS4A        MTC,CKB           0CE8
25     * STS4         CKST,MON,CCDS,EXM,RFB      0CE9
26     * STS5         AZ,MTRK,MTON,BRK,RFS,MST,MTZ 0CEA
27     * BRKI         BREAK,MONITOR FLAG      0CEB
28     * VAC          RANGE/FUNCTION TABLE      0CEC
29     * VDC          0CED
30     * RES          0CEE
31     * CAPA         0CEF
32     * DNFLG        DONE FLAG          0CF0
33     * DSPLA        KEYBOARD DISPLAY LSB      0CF1
34     * DSPMS        KEYBOARD DISPLAY MSB      0CF2
35     *
36     *      INPUT PORT ASSIGNMENTS:
37     * 0            KEYBOARD KEYS 7-0
38     * 01           KEYBOARD KEYS F-0
39     * 03           MRVK,MTRK,BRKK,EOC,AZK,
40     * 02           MSD,OVR,POL,MONK,FLT,
41     * 05           MTOT,MTOU
42     * 04           RGFN,MTCK
43     * 07           BTCK,BUCK
44     * 06           OTCK,OUCK
45     *
46     *      OUTPUT PORT ASSIGNMENTS:
47     * 00           DISPLAY VALUES
48     * 01           DISPLAY POSITIONS
49     * 05           AZ,,MTRV,MTON,BRK,RFS,MST,MTZ
50     * 07           MTZ,CKB
51     * 06           CKST,MON,CCSD,EXM,RFB
52     *
53     *      STACK SPACE NEEDED: 16 BYTES
54     *      MEMORY USED: 0CD0-0CF2

```

LINE LOC B1 B2 B3

SOURCE LINE

```

51      *          ALGORITHM SOURCE: C1C JOHN B. MCCORMACK
52      *          USAFA, CO 80840
53      *          PERM ADDRESS: 116 MILBURN LN
54      *          E. HILLS, NY 11577
55      *          (516) 621-1327
56      *          SOME OF THE SOFTWARE WAS BORROWED FROM
57      *          THE EDUMICRO MONITOR WRITTEN FOR USAFA/DFEE
58      *          (7 APRIL 78) BY ALAN J. LARSON.
59      *          WRITTEN: JULY, 1978
60      *          ****
61      *          ORG '0400
62      *          STPTR EQU '0CDF
63      *          BAY EQU '0CE0
64      *          TPBAY EQU '0CE1
65      *          CKT EQU '0CE2
66      *          TPCKT EQU '0CE3
67      *          MOT EQU '0CE4
68      *          PNTR EQU '0CE5
69      *          EC EQU '0CE7
70      *          STS4A EQU '0CE8
71      *          STS4 EQU '0CE9
72      *          STS5 EQU '0CEA
73      *          BRKI EQU '0CEB
74      *          VAC EQU '0CEC
75      *          VDC EQU '0CED
76      *          RES EQU '0CEE
77      *          CAPA EQU '0CEF
78      *          DNFLG EQU '0CF0
79      *          DSPLA EQU '0CF1
80      *          DSPMS EQU '0CF2
81      *          DN EQU 'F0
82      *          TELLS PNTR WHERE END OF TABLE
83      *          IS
84      *          ****
85      *          INIT
86      *          * PRESETS ALL STATUS REGISTERS, READS IN BAYS AND
87      *          * CIRCUITS TO BE TESTED FROM KEYBOARD, DECISION TO
88      *          * MONITOR, OR BREAK, AND ZERO TESTS THE LATS UNIT.
89      *          ****
90      0400 31 DF 0C  INIT LXI SP, STPTR
91      0403 21 EC 0C      LXI H, VAC
92      0406 22 E5 0C      SHLD PNTR          SET PNTR TO VAC
93      0409 21 E7 0C      LXI H, EC          SET EC TO ZERO
94      040C 36 00          MVI M, '00
95      040E 23            INX H
96      *          ADVANCE MEMORY TO SET
97      040F 36 00          *          MVI M, '00          OTHER STATUS WORDS
98      0411 23            INX H          STS4A, MTC AND CKB
99      0412 36 8D          MVI M, '8D          STS4, CKST(1), MON(0)
100     0414 23            INX H          CCSD(1), EXM(1), AND RFB(D)
          STS5, AZ(1), MTRV(0), MTON(0)

```

LINE	LOC	B1 B2 B3	SOURCE LINE	
101	0415	36 0F	MVI M, '0F	BRK(0),RFS(1),MST(1),MTZ(1)
102	0417	23	INX H	SKIP A PLACE IN MEMORY FOR B
103	0418	23	INX H	NEXT 4 LOCATIONS,VAC,VDC
104	0419	36 FD	MVI M, 'FD	RES AND CAPA ARE A TABLE
105	041B	23	INX H	OF RANGE FUNCTIONS THAT
106	041C	36 FB	MVI M, 'FB	BE POINTED TO BY THE
107	041E	23	INX H	POINTER, AND SCANNED IN
108	041F	36 F6	MVI M, 'F6	THE FNSCN SUBROUTINE
109	0421	23	INX H	
110	0422	36 F2	MVI M, 'F2	
111	0424	23	INX H	SET DNFLG TO NOT ZERO
112	0425	36 FF	MVI M, 'FF	
113	0427	CD 34 04	CALL INBAY	USER DETERMINES THE SCOPE
114	042A	CD C3 07	CALL RD2A	KEYIN 0 FOR BREAK,
115	042D	32 E0 0C	STA BRKI	BREAK FLAG STATUS WORD
116	0430	CD 60 04	CALL ZTEST	CHECK METER OPERATION
117	0433	C9	RET	
118				
119				
120				
121				
122				
123				
124				
125				
126				
127	0434	CD C3 07	INBAY CALL RD2A	READ IN LOWER BOUND OF BAYS
128	0437	32 E0 0C	STA BAY	SAVE IN STATUS WORD BAY
129	043A	CD C3 07	CALL RD2A	UPPER BOUND READ IN
130	043D	32 E1 0C	STA TPBAY	SAVE IN STATUS WORD TPBAY
131	0440	47	MOV B,A	
132	0441	3A E0 0C	LDA BAY	FETCH BAY
133	0444	B8	CMP B	
134	0445	CA 53 04	JZ INCKT	IF TESTING WITH A SINGLE
135			*	BAY, DECIDE WHICH CIRCUITS
136	0448	3E 00	MVI A, '00	IF MORE THAN 1 BAY, USE THE
137	044A	32 E2 0C	STA CKT	DEFAULT VALUES,00-79
138	044D	3E 79	DFCKT MVI A, '79	DEFAULT FOR TPCKT OF 79
139	044F	32 E3 0C	STA TPCKT	
140	0452	C9	RET	RETURN TO INIT
141	0453	CD C3 07	INCKT CALL RD2A	READ IN LOWER BOUND OF CIRCUIT
142	0456	32 E2 0C	STA CKT	SAVE IN STATUS WORD CKT
143	0459	CD C3 07	CALL RD2A	READ IN UPPER BOUND
144	045C	32 E3 0C	STA TPCKT	SAVE IN TPCKT
145	045F	C9	RET	RETURN TO INIT
146				
147				
148				
149				
150				

* INBAY READS IN UPPER AND LOWER LIMITS OF BAYS TO
 * BE TESTED AND STORES IN TWO STATUS WORDS IN RAM.
 * IF ONLY TESTING IN ONE BAY, IT ALSO INPUTS THE
 * DESIRED CIRCUITS TO BE TESTED. IF MORE THAN ONE
 * BAY HAS BEEN ENTERED, A DEFAULT VALUE OF 00
 * CIRCUITS IS USED.
 * THESE INPUTS SHOULD BE BCD.

INBAY CALL RD2A READ IN LOWER BOUND OF BAYS
 STA BAY SAVE IN STATUS WORD BAY
 CALL RD2A UPPER BOUND READ IN
 STA TPBAY SAVE IN STATUS WORD TPBAY
 MOV B,A
 LDA BAY FETCH BAY
 CMP B
 JZ INCKT IF TESTING WITH A SINGLE
 * BAY, DECIDE WHICH CIRCUITS
 IF MORE THAN 1 BAY, USE THE
 MVI A, '00 DEFAULT VALUES,00-79
 STA CKT DEFAULT FOR TPCKT OF 79
 DFCKT MVI A, '79
 STA TPCKT
 RET RETURN TO INIT
 INCKT CALL RD2A READ IN LOWER BOUND OF CIRCUIT
 STA CKT SAVE IN STATUS WORD CKT
 CALL RD2A READ IN UPPER BOUND
 STA TPCKT SAVE IN TPCKT
 RET RETURN TO INIT

* ZTEST ZERO TESTS THE LATS UNIT USING EVERY RANGE
 * AND FUNCTION FOR A SINGLE PAIR OF WIRES ON THE
 * FIRST CIRCUIT IN THE FIRST BAY, AND TESTS THE
 * UNIT FOR PROPER PERFORMANCE.

LINE	LOC	B1	B2	B3	SOURCE LINE
151					*****
152	0460	CD	E7	04	ZTEST CALL CKRT SELECT 1ST CIRCUIT
153	0463	CD	CF	05	CALL BRKMN EITHER BREAK OR MONITOR
154	0466	3A	EA	0C	LDA STS5 GET STS5 FOR MTZ BIT
155	0469	E6	FD		ANI 'FD CHANGE MTZ TO 0
156	046B	D3	05		OUT '05 MTZ TO LATS
157	046D	06	32		MVI B,'32 DELAY 50 MS TO GET VALID
158					SIGNAL
159	046F	32	EA	0C	* STA STS5 SAVE UPDATED STS5
160	0472	3A	E9	0C	LDA STS4 GET STS4 FOR RFB
161	0475	F6	0F		ORI '0F MAKE RFB 1111 (199 VAC)
162	0477	32	E9	0C	STA STS4 UPDATE STS4
163	047A	CD	83	06	CALL ZFNCT TEST METER CALIBRATION
164	047D	3A	EA	0C	LDA STS5 GET STS5 FOR MTZ
165	0480	F6	02		ORI '02 RESET MTZ BIT TO 1
166	0482	D3	05		OUT '05 CLEAR LATS
167	0484	06	32		MVI B,'32 DELAY 50 MS
168	0486	CD	DF	07	CALL DBMS
169	0489	32	EA	0C	STA STS5 UPDATE STS5
170	048C	C9			RET RETURN TO INIT
171					*****
172					* RGFN EXPECTS THE PROPER STATUS WORD IN THE A REG.
173					* IT SAVES STS4, OUTPUTS THE RANGE AND FUNCTION TO
174					* THE LATS, AND INSURES THE INFORMATION WAS RECEIVED
175					*****
176	048D	32	E9	0C	RGFN STA STS4 SAVE STS4
177	0490	D3	06		OUT '06 PUT RFB ON LINES
178					* (MUST STILL STROBE RFS)
179	0492	5F			MOV E,A SAVE STS4 IN E
180	0493	3E	00		MVI A,'00 INITIALIZE EC
181	0495	32	E7	0C	STA EC
182	0498	3A	EA	0C	LDA STS5 GET STS5 FOR RFS
183	049B	E6	F7		ANI 'F7 BRING RFS LOW
184	049D	D3	05		OUT '05 STROBE RFS TO ENTER RFB
185	049F	06	64		MVI B,'64 DELAY 100 MS
186	04A1	CD	DF	07	CALL DBMS
187	04A4	F6	08		ORI '08 TAKE RFS BACK UP TO 1
188	04A6	D3	05		OUT '05
189	04A8	7B			MOV A,E CHECK RFB FOR CAP OR AC
190	04A9	E6	0F		ANI '0F MASK RFB FROM STS4
191	04AB	57			MOV D,A SAVE RFB IN D REGISTER
192	04AC	FE	02		CPI '02
193	04AE	CA	DA	04	JZ LONG CAPA AND VAC NEED MORE DELAY
194	04B1	FE	03		CPI '03
195	04B3	CA	DA	04	JZ LONG
196	04B6	FE	0F		CPI '0F
197	04B8	CA	DA	04	JZ LONG
198	04BB	FE	0E		CPI '0E
199	04BD	CA	DA	04	JZ LONG
200	04C0	FE	0D		CPI '0D

LINE	LOC	B1 B2 B3	SOURCE LINE
201	04C2	CA DA 04	JZ LONG
202	04C5	06 0A	MVI B,'0A
203	04C7	CD DF 07	CALL DBMS
204	04CA	DB 04	IN '04
205	04CC	E6 F0	ANI 'F0
206	04CE	0F	RRC
207	04CF	0F	RRC
208	04D0	0F	RRC
209	04D1	0F	RRC
210	04D2	BA	CMP D
211	04D3	C8	RZ
212	04D4	CD 74 05	CALL ECHO
213	04D7	C3 8D 04	JMP RGFN
214	04DA	06 FA	MVI B,'FA
215	04DC	CD DF 07	CALL DBMS
216	04DF	06 FA	MVI B,'FA
217	04E1	CD DF 07	CALL DBMS
218	04E4	C3 CA 04	JMP CHK
219			
220			
221			
222			
223	04E7	3E 00	CKRT MVI A,00
224	04E9	32 E7 0C	STA EC
225	04EC	3A EA 0C	CRT1 LDA STS5
226	04EF	E6 EF	ANI 'EF
227	04F1	D3 05	OUT '05
228	04F3	32 EA 0C	STA STS5
229	04F6	3A EA 0C	LDA STS4
230	04F9	E6 8F	ANI '8F
231	04FB	D3 06	OUT '06
232	04FD	06 C9	MVI B,'C9
233	04FF	CD DF 07	CALL DBMS
234	0502	32 E9 0C	STA STS4
235	0505	DB 02	IN '02
236	0507	E6 40	ANI '40
237	0509	C2 1A 05	JNZ ER
238	050C	DB 03	IN '03
239	050E	E6 20	ANI '20
240	0510	C2 1A 05	JNZ ER
241	0513	CD 59 05	CALL STROB
242	0516	CD AC 05	CALL CHECK
243	0519	C9	RET
244	051A	CD 74 05	ER CALL ECHO
245	051D	C3 EC 04	JMP CRT1
246			
247			
248			
249			
250			

DELAY VDC AND RES 10 MS
 INPUT FOR RFN
 MASK FOR RFN
 PUT IN POSITION TO CHECK
 RFB=RFN?
 RETURN IF RFB=RFN
 ERROR ROUTINE IF RFB NEQ RFN
 IF NOT MANY ERRORS, TRY AGAIN
 DELAY 500 MS IF CAP OR VAC
 * CKRT OUTPUTS THE DESIRED CIRCUIT TO LATS, AND
 * CHECKS TO INSURE THE INFORMATION WAS RECEIVED.
 * CKCNT COUNTS THROUGH THE CIRCUITS, AND IS CALLED
 * WHEN ALL PAIRS HAVE BEEN TESTED IN THE PREVIOUS
 * CIRCUIT. IT CALLS BYCNT WHEN ALL THE CIRCUITS
 * IN A BAY HAVE BEEN TESTED.

LINE	LOC	B1 B2 B3	SOURCE LINE
251			*****
252	0520	3A F0 0C	CKCNT LDA DNFLG CHECK FLAG FOR ALL DONE
253	0523	FE 00	CPI '00 IF DONE, JUMP OUT OF ROUTINE
254	0525	CA EA 07	JZ DONE
255	0528	CD E7 04	CALL CKRT ACCESS DESIRED CIRCUIT
256	052B	3A E3 0C	LDA TPCKT GET TPCKT
257	052E	47	MOV B,A
258	052F	3A E2 0C	LDA CKT GET CKT
259	0532	B8	CMP B ALL CIRCUITS TESTED?
260	0533	CC 3E 05	CZ BYCNT YES? NEXT BAY
261	0536	3C	INR A INCREASE CIRCUIT COUNT
262	0537	37	STC IN BCD. CARRY BIT
263	0538	3F	CMC MUST BE ZERO FOR DAA TO
264	0539	27	DAA WORK.
265	053A	32 E2 0C	STA CKT RETURN TO CKT
266	053D	C9	RET
267			*****
268			* BYCNT IS VERY SIMILAR TO CKCNT, BUT INSTEAD OF
269			* CIRCUITS. IT COUNTS BAYS. IT HALTS WHEN ALL BAYS
270			* HAVE BEEN TESTED.
271			*****
272	053E	3A E1 0C	BYCNT LDA TPBAY GET TPBAY
273	0541	47	MOV B,A
274	0542	3A E0 0C	LDA BAY GET BAY
275	0545	B8	CMP B BAY = TPBAY?
276	0546	C2 4F 05	JNZ BINC
277	0549	3E 00	MVI A, '00 YES? ALL TESTING DONE
278	054B	32 F0 0C	STA DNFLG SO SET DNFLG TO ZERO
279	054E	C9	RET
280	054F	3C	BINC INR A
281	0550	37	STC
282	0551	3F	CMC
283	0552	27	DAA
284	0553	32 E0 0C	STA BAY UPDATE BAY
285	0556	3E 99	MVI A, '99 RESET CKT BEFORE RET TO CKCNT
286	0558	C9	RET
287			*****
288			* STROBE SEES THAT THE BAYS AND CIRCUITS ARE TRANS-
289			* MITTED TO LATS PROPERLY. OUTPT IS THE SUBROUTINE
290			* THAT DOES THE BULK OF THE WORK.
291			*****
292	0559	3A E9 0C	STROB LDA STS4 GET CCSD
293	055C	E6 DF	ANI 'DF TAKE LOW TO CLEAR LATS
294	055E	D3 06	OUT '06
295	0560	CD 4C 07	CALL HUNUS
296	0563	F6 20	ORI '20 TAKE BACK HIGH AFTER STROBE
297	0565	D3 06	OUT '06
298	0567	3A E0 0C	LDA BAY FETCH BAY BAY=INF
299	056A	CD 83 05	CALL OUTPT
300	056D	3A E2 0C	LDA CKT FETCH CKT CKT=INF

LINE	LOC	B1 B2 B3	SOURCE LINE
301	0570	CD 83 05	CALL OUTPT
302	0573	C9	RET
303			*****
304			* ECHO IS A ROUTINE CALLED WHENEVER THERE HAS BEEN
305			* AN ECHO ERROR, IE, DIFFERENT INFORMATION BACK FROM
306			* LATS THAT WAS GIVEN TO IT. ECHO PROVIDES A LITTLE
307			* PADDING BY REPEATING THE TRANSFER A FEW TIMES
308			* BEFORE PRINTING THE ERROR MESSAGE.
309			*****
310	0574	3A E7 0C	ECHO LDA EC FETCH EC
311	0577	3C	INR A INCREASE IT
312	0578	FE 10	CPI '10 HOW MANY READ ERRORS?
313	057A	32 E7 0C	STA EC UPDATE EC
314	057D	C0	RNZ FEW ERRORS, RETURN AND
315			* TRY AGAIN
316	057E	1E 01	MVI E,'01 MANY ERRORS, GIVE ERROR
317	0580	C3 53 07	JMP ERMS MESSAGE 01
318			*****
319			* LATS EXPECTS BAY AND CIRCUIT INFORMATION IN A
320			* STRING OF 4 BIT WORDS STROBED BY CKST. EDUMICRO
321			* USES 8 BIT WORDS WHICH MUST BE SHIFTED AND MASKED
322			* BEFORE TRANSMISSION. OUTPT DOES THIS. WHILE STROB
323			* KEEPS TRACK OF WORDS.
324			*****
325	0583	1E 00	OUTPT MVI E,'00 INITIALIZE N
326	0585	57	MOV D,A
327	0586	7B	BACK MOV A,E
328	0587	FE 02	CPI '02 N ITERATIONS?
329	0589	C8	RZ
330	058A	7A	MOV A,D INF
331	058B	0F	RRC MOVE TENS DIGIT TO RIGHT
332	058C	0F	RRC FIRST TIME THRU, AND UNITS
333	058D	0F	RRC TO RIGHT SECOND TIME THRU
334	058E	0F	RRC
335	058F	57	MOV D,A
336	0590	3A E8 0C	LDA STS4A GET STS4A FOR CKB
337	0593	47	MOV B,A
338	0594	7A	MOV A,D
339	0595	E6 0F	ANI '0F MASK OFF EITHER TENS OR UNITS
340	0597	B0	ORA B
341	0598	D3 07	OUT '07 OUTPUT STS4A
342	059A	3A E9 0C	LDA STS4 GET STS4 FOR CKST
343	059D	E6 7F	ANI '7F BRING CKST LOW
344	059F	D3 06	OUT '06
345	05A1	CD 4C 07	CALL HUNUS HOLD LOW FOR 100US
346	05A4	F6 00	ORI '00
347	05A6	D3 06	OUT '06
348	05A8	1C	INR E INCREASE N
349	05A9	C3 86 05	JMP BACK SECOND HALF OF WORD
350			*****

LINE	LOC	B1	B2	B3	SOURCE LINE
351					* CHECK MAKES SURE THAT THE CORRECT CIRCUIT AND BAY
352					* WAS RECEIVED BY LATS.
353					*****
354	05AC	06	32		CHECK MVI B,'32 DELAY 50 MS FOR BCK AND OCK
355	05AE	CD	DF	07	CALL DBMS TO SETTLE
356	05B1	DB	07		IN '07 GET BCK
357	05B3	47			MOV B,A
358	05B4	3A	E0	0C	LDA BAY FETCH BAY
359	05B7	B8			CMP B BAY=BYCK?
360	05B8	CA	C4	05	JZ CKCK
361	05B8	CD	74	05	ERR CALL ECHO BAY NEQ BYCK,OCK NEQ CKT,
362	05B8	CD	59	05	CALL STROB REPEAT INPUT
363	05C1	C3	AC	05	JMP CHECK
364	05C4	DB	06		CKCK IN '06 GET OCK
365	05C6	47			MOV B,A
366	05C7	3A	E2	0C	LDA CKT FETCH CKT
367	05CA	B8			CMP B OCK=CKT?
368	05CB	C2	BB	05	JNZ ERR
369	05CE	C9			RET
370					*****
371					* BREAKS OR MONITORS CIRCUIT BASED ON STATUS WORD
372					* BRKI,0 FOR BREAK, ALL OTHER FOR MONITOR
373					*****
374	05CF	3E	00		BRKMN MVI A,00
375	05D1	32	E7	0C	STA EC INITIALIZE EC
376	05D4	3A	EB	0C	AA LDA BRKI GET BRKI
377	05D7	F6	00		ORI '00 SET FLAGS
378	05D9	CA	F9	05	JZ BKST IF BRKI=0,BREAK CIRCUIT
379	05DC	3A	E9	0C	LDA STS4 GET STS4 FOR MON
380	05DF	F6	40		ORI '40 SET MON=1
381	05E1	D3	06		OUT '06
382	05E3	06	C9		MVI B,'C9
383	05E5	CD	DF	07	CALL DBMS DELAY 200 MS FOR VALID MON
384	05E8	32	E9	0C	STA STS4 RETURN STS4
385	05EB	CD	54	06	CALL FLT CHECK FOR FLT
386	05EE	DB	02		IN '02 CHECK MONK
387	05F0	E6	40		ANI '40
388	05F2	C0			RNZ MONK RECEIVED, RETURN
389	05F3	CD	74	05	CALL ECHO NOT RECEIVED? ERROR ROUTINE
390	05F6	C3	D4	05	JMP AA
391	05F9	3A	EA	0C	BKST LDA STS5 GET STS5 FOR BRK
392	05FC	F6	10		ORI '10 SET BRK HIGH TO BREAK
393	05FE	D3	05		OUT '05
394	0600	06	C9		MVI B,'C9 DELAY 200 MS FOR VALID BRK
395	0602	CD	DF	07	CALL DBMS
396	0605	32	EA	0C	STA STS5 SAVE STS5
397	0608	CD	54	06	CALL FLT CHECK FOR FLT
398	060B	DB	03		IN '03 BRKK
399	060D	E6	20		ANI '20
400	060F	C0			RNZ BRKK RECEIVED, RETURN

LINE	LOC	B1 B2 B3	SOURCE LINE
401	0610	CD 74 05	CALL ECHO
402	0613	C3 D4 05	JMP AA
403			*****
404			* COUNTS THROUGH MATRIX OF PAIRS OF WIRES, AND
405			* GROUNDS. ONCE ALL PAIRS TESTED, IT CALLS CKCNT
406			*****
407	0616	3A E8 0C	PRCNT LDA STS4A GET STS4A FOR MTC
408	0619	FE A0	CPI 'A0 ALL PAIRS TESTED
409	061B	CA 27 06	JZ CIR YES? GO TO NEXT CIRCUIT
410	061E	CD 35 06	PRCL CALL PR
411	0621	C6 10	ADI '10 INCREASE MTC
412	0623	32 E8 0C	STA STS4A RESTORE STS4A
413	0626	C9	RET
414	0627	CD 20 05	CIR CALL CKCNT SET UP NEXT CIRCUIT
415	062A	CD CF 05	CALL BRKMN AND MONITOR OR BREAK IT
416	062D	3E 00	MVI A, '00 INITIALIZE MTC
417	062F	32 E8 0C	STA STS4A UPDATE STS4A
418	0632	C3 1E 06	JMP PRCL AND EXAMINE THE FIRST PAIR
419			*****
420			* EXPECTS PAIR TO BE CALLED IN STS4A, THIS ROUTINE
421			* OUTPUTS THE MATRIX TO LATS AND CHECKS FOR INFO
422			* RECEIVED.
423			*****
424	0635	57	PR MOV D,A DO NOT DESTROY STS4A
425	0636	3E 00	MVI A,00 INITIALIZE EC
426	0638	32 E7 0C	STA EC
427	063B	7A	PR1 MOV A,D GET STS4A
428	063C	D3 07	OUT '07 OUT TO LATS
429	063E	06 0A	MVI B, '0A
430	0640	CD DF 07	CALL DBMS DELAY 10 MS FOR VALID SIGNAL
431	0643	47	MOV B,A
432	0644	DB 04	IN '04 MTC INPUT
433	0646	E6 0F	ANI '0F MASK OFF MTCK
434	0648	0F	RRC ROTATE TO MTC POSITION
435	0649	0F	RRC
436	064A	0F	RRC
437	064B	0F	RRC
438	064C	B8	CMP B MTCK = MTC?
439	064D	C8	RZ
440	064E	CD 74 05	CALL ECHO
441	0651	C3 3B 06	JMP PR1 NOT EQUAL, REPEAT
442			*****
443			* CHECKS FOR FLT BIT AFTER BRK OR MON, IF FLT, GOES
444			* TO ERROR MESSAGE.
445			*****
446	0654	DB 02	FLT IN '02 INPUT AND MASK FLT BIT
447	0656	E6 00	ANI '00 (ACTIVE LOW)
448	0658	C0	RNZ
449	0659	1E 02	MVI E, '02 ERROR MESSAGE 02, FLT SIGNAL
450	065B	C3 53 07	JMP ERMS DISPLAY ERROR MESSAGE

LINE	LOC	B1	B2	B3	SOURCE LINE
451					*****
452					* READ TAKES THE READING OFF THE METER AND STORES IT
453					* IN MOT STATUS WORD. OVR BIT SAYS METER OFF SCALE,
454					* READ CHECKS AND CHANGES SCALE. IF TOP SCALE, ERROR
455					* MESSAGE IS GENERATED.
456					*****
457	065E	3E	00		READ MVI A,00 INITIALIZE EC
458	0660	32	E7	0C	STA EC
459	0663	DB	03		RD1 IN '03 CHECK MTON (ACTIVE HIGH)
460	0665	E6	40		ANI '40
461	0667	CA	6E	06	JZ ONMT IF NOT, TURN METER ON
462	066A	CD	A1	06	CALL DREAD IF ON, READ METER
463	066D	C9			RET
464	066E	3A	EA	0C	ONMT LDA STS5 NOT ON, GET STS5 FOR MTON
465	0671	F6	20		ORI '20 MTON HIGH, TURN METER ON
466	0673	32	EA	0C	STA STS5 SAVE STS5
467	0676	D3	05		OUT '05
468	0678	06	32		MVI B, '32 DELAY 50 MS FOR VALID SIGNAL
469	067A	CD	DF	07	CALL DBMS
470	067D	CD	74	05	CALL ECHO
471	0680	C3	63	06	JMP RD1
472					*****
473					* COUNTS THROUGH FUNCTIONS AND RANGES FOR ZERO TEST
474					*****
475	0683	CD	8D	04	ZFNCT CALL RGFN OUTPUT RFB TO LATS
476	0686	CD	8B	06	CALL ZREAD CHECK VALUES FROM METER
477	0689	3A	E9	0C	BB LDA STS4 GET STS4 FOR RFB
478	068C	3D			DCR A MOVE TO NEXT FUNCTION (IN
479					* TEST, MUST CHECK ALL RANGES
480					* AND FUNCTIONS
481	068D	4F			MOV C,A SAVE NEW STS4
482	068E	E6	0F		ANI '0F MASK RFB
483	0690	FE	05		CPI '05 THIS RANGE NOT POSSIBLE
484	0692	CA	89	06	JZ BB SO DECREASE AGAIN
485	0695	FE	04		CPI '04 SAME WITH RANGE 04
486	0697	CA	89	06	JZ BB
487	069A	FE	01		CPI '01 CHECK ALL DONE
488	069C	79			MOV A,C STS4 MUST BE IN A FOR RGFN
489	069D	C8			RZ
490	069E	C3	83	06	JMP ZFNCT REPEAT LOOP UNTIL ALL DONE
491					*****
492					* DREAD STROBES METER, WAITS FOR IT TO SETTLE, AND
493					* CHECKS OVR BIT, CALLED FROM ZREAD OR MREAD.
494					*****
495	06A1	3A	EA	0C	DREAD LDA STS5 GET STS5 FOR MST
496	06A4	E6	FB		ANI 'FB BRING MST LOW
497	06A6	D3	05		OUT '05
498	06A8	CD	4C	07	CALL HUNUS STROBE MST LOW FOR 100 US
499	06AB	F6	04		ORI '04 TAKE MST HIGH
500	06AD	D3	05		OUT '05 END OF PULSE

LINE	LOC	B1	B2	B3	SOURCE LINE	
501	06AF	DB	03		WAIT IN '03	LOOK FOR EOC (END OF
502					*	CONVERSION)
503	06B1	E6	10		ANI '10	MASK EOC AND WAIT FOR LOW
504	06B3	C2	AF	06	JNZ WAIT	
505	06B6	DB	02		IN '02	LOOK FOR OVR
506	06B8	E6	10		ANI '10	AND MASK
507	06BA	C9			RET	
508						
509					*****	
510					* ZREAD CHECKS TOLLERENCES DURING ZERO TESTING, ALL	
511					* READINGS MUST BE WITHIN +-5, AND CAPA SHOULD OVER-	
512					* FLOW.	
513	06BB	CD	5E	06	ZREAD CALL READ	WAIT FOR READING, CHECK OVR
514	06BE	CA	D2	06	JZ CAP	EXPECT OVR IF CAP
515	06C1	CD	E0	06	CALL CKRFN	CHECK RFB FOR CAP
516	06C4	F6	00		ORI '00	SET FLAGS
517	06C6	C4	DB	06	CNZ ERZ	IF CAP, AND NOT OVR,
518					*	ZERO TEST FAILS
519	06C9	DB	05		IN '05	READ IN MTO
520	06CB	47			MOV B,A	
521	06CC	DE	05		SBI '05	
522	06CE	F4	DB	06	CP ERZ	IF MTO >5, ZERO TEST FAILED
523	06D1	C9			RET	
524	06D2	CD	E0	06	CAP CALL CKRFN	CHECK RFB FOR OTHER THAN CAP
525	06D5	F6	00		ORI '00	ONLY CAP SHOULD OVERFLOW
526	06D7	CC	DB	06	CZ ERZ	
527	06DA	C9			RET	
528	06DB	1E	03		ERZ MVI E,03	ERROR MESSAGE 03, ZERO TEST
529					*	FAILURE
530	06DD	C3	53	07	JMP ERMS	DISPLAY MESSAGE
531	06E0	3A	E9	0C	CKRFN LDA STS4	STS4 FOR RFB
532	06E3	E6	0F		ANI '0F	MASK RFB
533	06E5	FE	02		CPI '02	CHECK FOR CAP
534	06E7	CA	F2	06	JZ NG	IF CAP, SET CAPFLG TO A 1
535	06EA	FE	03		CPI '03	(OTHER RFB FOR CAP)
536	06EC	CA	F2	06	JZ NG	
537	06EF	3E	00		MVI A,00	SET CAPFLG TO A 0
538	06F1	C9			RET	
539	06F2	3E	01		NG MVI A,01	
540	06F4	C9			RET	
541					*****	
542					*MREAD CHECKS AND INCREMENTS RANGE ON AN OVERFLOW	
543					*AND STORES METER READING IN MOT.	
544					*****	
545	06F5	CD	5E	06	MREAD CALL READ	METER READ
546	06F8	CA	01	07	JZ CRNG	CHANGE RANGE IF OVR
547	06FB	DB	05		IN '05	INPUT MOT
548	06FD	32	E4	0C	STA MOT	SAVE MOT
549	0700	C9			RET	MOT IS IN RAM FOR OPERATOR
550					*****	

LINE	LOC	B1	B2	B3	SOURCE LINE
551					* CHANGES RANGE IF OVR AND NOT ON TOP SCALE, IF TOP
552					* SCALE, GIVES ERROR MESSAGE.
553					*****
554	0701	3A	E9	0C	CRNG LDA STS4 GET STS4 FOR RFB
555	0704	47			MOV B,A
556	0705	E6	0F		ANI '0F MASK RFB
557	0707	FE	0F		CPI '0F MAKE SURE METER IS NOT AT TOP
558	0709	CC	28	07	CZ ERM RANGE FOR VAC,VDC,RES OR CAP
559	070C	FE	0C		CPI '0C
560	070E	CC	28	07	CZ ERM
561	0711	FE	0A		CPI '0A
562	0713	CC	28	07	CZ ERM
563	0716	FE	03		CPI '03
564	0718	CC	28	07	CZ ERM
565	071B	3C			INR A GO TO NEXT RANGE
566	071C	4F			MOV C,A
567	071D	78			MOV A,B RETRIEVE OLD STS4
568	071E	E6	F0		ANI 'F0 TO GET EXM,CCDS,MON,CKST
569	0720	B1			ORA C
570	0721	CD	8D	04	CALL RGFN NEW STS4 TO LATS
571	0724	CD	5E	06	CALL READ TRY INPUT METER UNDER NEW
572					* RANGE (MUST CALL TO KEEP
573					* STACK STRAIGHT).
574	0727	C9			RET
575	0728	1E	04		ERM MVI E,'04 ERROR MESSAGE 04, OVER RANGE
576	072A	C3	53	07	JMP ERMS DISPLAY ERROR MESSAGE
577					*****
578					* FNCSN USES THE POINTER IN RAM TO SCAN THE FOUR
579					* FUNCTIONS LATS CAN MEASURE, RESETS THE POINTER, AND
580					* GOES TO THE NEXT PAIR WHEN ALL FUNCTIONS TESTED.
581					*****
582	072D	2A	E5	0C	FNCSN LHLD PNTR GET PNTR
583	0730	7D			MOV A,L
584	0731	FE	F0		CPI DN ALL FUNCTIONS DONE?
585	0733	CC	45	07	CZ PAIR IF ALL DONE, GO TO NEXT PAIR
586	0736	46			MOV B,M PUT RFB IN B
587	0737	3A	E9	0C	LDA STS4 GET STS4 FOR OLD RFB
588	073A	F6	0F		ORI '0F MASK STS4 OF OLD RFB
589	073C	A0			ANA B UPDATE STS4
590	073D	CD	8D	04	CALL RGFN SAVE STS4 AND OUTPUT RANGE
591	0740	23			INX H
592	0741	22	E5	0C	SHLD PNTR ADVANCE AND SAVE POINTER
593	0744	C9			RET
594	0745	CD	16	06	PAIR CALL PRCNT GO TO NEXT PAIR
595	0748	21	EC	0C	LXI H,VAC INITIALIZE POINTER
596	074B	C9			RET
597					*****100 U SECOND DELAY*****
598	074C	06	04		HUNUS MVI B,04 DELAY OF 100 US FOR STROBING
599	074E	05			ONE DCR B
600	074F	C2	4E	07	JNZ ONE

LINE	LOC	B1 B2 B3	SOURCE LINE
601	0752	C9	RET
602			*****
603			* PRINT ERROR MESSAGE * 01 TRANSFER ERROR
604			* 02 FLT SIGNAL
605			* 03 ZERO TEST FAILURE
606			* 04 OVERFLOW
607			* WAIT FOR KEYIN BEFORE DISPLAYING RETURN ADDRESS
608			* AND JUMP TO ADDRESS KEYED IN
609			*****
610	0753	CD 75 07	ERMS CALL DS1BY DISPLAY EM
611	0756	CD 93 07	CALL KEYIN WAIT FOR KEY BEFORE DISPLAY
612	0759	D1	POP D GET RET ADDRESS OFF STACK
613	075A	CD 65 07	CALL DSDE DISPLAY RETURN ADDRESS
614	075D	CD D5 07	CALL RD2HL INPUT 2 BYTES INTO H-L
615	0760	E9	PCHL JUMP TO INPUTTED ADDRESS
616			*****
617			* ROUTINE DISPLAYS VARIOUS THINGS DEPENDING ON ENTRY
618			* POINT. DS2BY: CONTENTS OF 2 MEMORY LOCATIONS
619			* DSDE : CONTENTS OF D,E REGISTERS
620			* DS1BY: CONTENTS OF E REGISTER
621			* REGISTERS: ALL
622			* MEMORY: DISPLA,DSPMS
623			* STACK SPACE: 2 BYTES
624			* OUTPORTS: 0,1
625			* PROGRAMMER: AL LARSON
626			*****
627	0761	2A F1 0C	DS2BY LHLD DSPLA GET LOCATIONS OF DATA
628	0764	EB	XCHG PUT IN D,E
629	0765	7A	DSDE MOV A,D DISPLAY DIGIT 2
630	0766	06 04	MVI B,'04 LOCATION TO DISPLAY
631	0768	CD 02 07	CALL DISPL
632	076B	7A	MOV A,D DISPLAY DIGIT 3
633	076C	0F	RRC
634	076D	0F	RRC
635	076E	0F	RRC
636	076F	0F	RRC
637	0770	06 08	MVI B,'08
638	0772	CD 02 07	CALL DISPL
639	0775	7B	DS1BY MOV A,E DISPLAY DIGIT 0
640	0776	06 01	MVI B,'01
641	0778	CD 02 07	CALL DISPL
642	077B	7B	MOV A,E DISPLAY DIGIT 1
643	077C	0F	RRC
644	077D	0F	RRC
645	077E	0F	RRC
646	077F	0F	RRC
647	0780	06 02	MVI B,'02 PLCE DIGIT IN DISPLAY 1
648			*****
649			*DISPL DISPLAYS A(3:0) IN KEYBOARD DISPLAY LOCATIONS
650			*DESIGNATED BY REGISTER B. THE LSB OF B CORRESPONDS

LINE LOC B1 B2 B3

SOURCE LINE

```

551      *TO THE LSB OF THE DISPLAY LIGHTS ETC.A LOGIC 1 IN A
552      *BIT POSITION CAUSES A(3:0) TO BE DISPLAYED IN THE
553      *CORRESPONDING DISPLAY POSITION.A (3:0) MAY BE
554      *DISPLAYED IN UP TO 5 POSITIONS SIMULTANEOUSLY. 0 IN
555      *B CAUSES CORRESPONDING DISPLAY POSITION TO REMAIN
556      * REGISTERS: A,C
557      * OUTPORTS: 0,1
558      * PROGRAMMER: AL LARSON, MODIFIED BY M.VERSTEGEN
559      ****
560 0782 E6 0F      DISPL ANI '0F      RESET UPPER TO UNBLANK
561 0784 D3 01      OUT '01      SEND TO DISPLAY
562 0786 3E FF      MVI A,'FF      SEND LOAD COMMAND TO PORT 0
563 0788 D3 00      OUT '00
564 078A 78        MOV A,B      GET LOAD WORD
565 078B 2F        CMA          COMPLEMENT
566 078C D3 00      OUT '00      AND SEND
567 078E 3E FF      MVI A,'FF      RESEND ZERO VALUE
568 0790 D3 00      OUT '00
569 0792 C9        RET
570      ****
571      *KEYIN SENSES 1 KEY DOWN, DEBOUNCES, DECODES, AND PUTS
572      *HEX VALUE IN A REGISTER
573      * REGISTERS: A,B,C,D,E
574      * INPORTS: 0,1
575      * PROGRAMMER: AL LARSON
576      ****
577 0793 DB 00      KEYIN IN '00      READ KEYS 7-0
578 0795 2F        CMA          INVERT
579 0796 4F        MOV C,A      SAVE A COPY
580 0797 DB 01      IN '01      READ KEYS F-8
581 0799 2F        CMA
582 079A 47        MOV B,A
583 079B B1        ORA C      CHECK FOR ANY KEY PRESSED
584 079C CA 93 07   JZ KEYIN    NONE DOWN, WAIT IN THIS LOOP
585 079F 1E 00      KEYCD MVI E,00 INITIALIZE HEXCODE
586 07A1 79        MOV A,C      GET 7-0
587 07A2 16 08      LOOP MVI D,'08 SET UP 8 BIT COUNTER
588 07A4 0F        ROTAT RRC    PUT NEXT LS BIT IN CARRY
589 07A5 DA B1 07   JC KEYUP    DONE IF THIS WAS KEYDOWN
590 07A6 1C        INR E      ADVANCE HEXCODE
591 07A7 15        DCR D      SEE IF WORD IS DONE
592 07A9 C2 A4 07   JNZ ROTAT   NO? KEEP SHIFTING FOR KEY
593 07AD 78        MOV A,B      PUT KEYS F-8 IN A
594 07AE C3 A2 07   JMP LOOP    LOOK FOR KEYDOWN IN SECOND
595 07B1 16 00      KEYUP MVI D,'00 SET DELAY COUNT
596 07B3 DB 00      READU IN '00 READ KEYS 0-7
597 07B5 47        MOV B,A      SAVE IN B
598 07B6 DB 01      IN '01      READ KEYS 8-F
599 07B8 A0        ANA B      CHECK ALL UP
700 07B9 FE FF      CPI 'FF

```

LINE	LOC	B1 B2 B3	SOURCE LINE
701	078B	C2 B1 07	JNZ KEYUP KEY STILL DOWN, WAIT
702	078E	15	DCR D ALL UP, DELAY TO DEBOUNCE
703	078F	C2 B3 07	JNZ READU LOOP UNTIL DELAY DONE
704	07C2	C9	RET
705			*****
706			*RD2A AND RD2HL WERE ROUTINES WRITTEN FOR EDUMICRO O/S
707			*MODIFIED FOR THIS PROGRAM
708			* REGISTERS: ALL
709			* MEMORY: DSPLA,DSPMS
710			* STACK: 5 BYTES
711			* INPORTS: 0,1
712			* OUTPORTS:0,1
713			*****
714	07C3	CD 93 07	RD2A CALL KEYIN READ KEY CHARACTER TO A
715	07C6	0F	RRC SHIFT TO MS POSITION
716	07C7	0F	RRC
717	07C8	0F	RRC
718	07C9	0F	RRC
719	07CA	6F	MOV L,A SAVE KEYED CHARACTER
720	07CB	CD 93 07	CALL KEYIN READ LS KEYED CHARACTER
721	07CE	B5	ORA L
722	07CF	5D	MOV E,L
723	07D0	CD 75 07	CALL DS18Y
724	07D3	7B	MOV A,E EXPECT VALUE IN A
725	07D4	C9	RET
726	07D5	CD C3 07	RD2HL CALL RD2A
727	07D8	E5	PUSH H SAVE FIRST SET OF CHARACTERS
728	07D9	CD C3 07	CALL RD2A AND GET SECOND SET
729	07DC	E1	POP H AS WELL AS FIRST
730	07DD	67	MOV H,A SET SECOND IN H
731	07DE	C9	RET
732			*****
733			* DELAYS B MILLISECONDS WHERE B IS THE VALUE IN B
734			* REGISTERS: B,C
735			* PROGRAMMER: AL LARSON
736			*****
737	07DF	0E 3D	DBMS MVI C,'3D
738	07E1	0D	DIMS DCR C
739	07E2	C2 E1 07	JNZ DIMS
740	07E5	05	DCR B
741	07E6	C2 DF 07	JNZ DBMS
742	07E9	C9	RET
743	07EA	76	DONE HLT
744			END

AFCS 8080 ASSEMBLER

SYMBOL TABLE

SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE
STPTR	0C DF	BAY	0C E0	TPBAY	0C E1	CKT	0C E2
TPCKT	0C E3	MOT	0C E4	PNTR	0C E5	EC	0C E7
STS4A	0C E8	STS4	0C E9	STSS	0C EA	BRKI	0C EB
VAC	0C EC	VDC	0C ED	RES	0C EE	CAPA	0C EF
DNFLG	0C F0	DSPLA	0C F1	DSPMS	0C F2	DN	00 F0
INIT	04 00	INBAY	04 34	DFCKT	04 4D	INCKT	04 53
ZTEST	04 60	RGFN	04 8D	CHK	04 CA	LONG	04 DA
CKRT	04 E7	CRT1	04 EC	ER	05 1A	CKCNT	05 20
BYCNT	05 3E	BINC	05 4F	STROB	05 59	ECHO	05 74
OUTPT	05 83	BACK	05 86	CHECK	05 AC	ERR	05 8B
CKCK	05 C4	BRKMN	05 CF	AA	05 D4	BKST	05 F9
PRCNT	06 16	PRCL	06 1E	CIR	06 27	PR	06 35
PR1	06 3B	FLT	06 54	READ	06 5E	RD1	06 63
DNMT	06 6E	ZFNCT	06 83	BB	06 89	DREAD	06 A1
WAIT	06 AF	ZREAD	06 8B	CAP	06 D2	ERZ	06 DB
CKRFN	06 E0	NG	06 F2	MREAD	06 F5	CRNG	07 01
ERM	07 28	FNSCN	07 2D	PAIR	07 45	HUNUS	07 4C
ONE	07 4E	ERMS	07 53	DS2BY	07 61	DSDE	07 65
DS1CY	07 75	DISPL	07 82	KEYIN	07 83	KEYCD	07 8F
LOOP	07 A2	ROTAT	07 A4	KEYUP	07 B1	READU	07 B3
R22A	07 C3	RD2HL	07 D5	DBMS	07 DF	DIMS	07 E1
DONE	07 EA						

** ASSEMBLY COMPLETE, NO ERRORS 1003 BYTES ASSEMBLED **